In the Specification:

PLEASE MAKE THE FOLLOWING CORRECTIONS:

Paragraph [0005] beginning on page 2:

[0005] Currently, correlations of the experimental data with types of additional information as exemplified above are often done by manually (i.e., visually) inspecting the additional (e.g., clinical) data and visually comparing it with the experimental data to look for similarities (i.e., correlations) between experimental and observed phenomena. For example, a researcher might notice a highly up or down regulated gene during inspection of a microarray experiment and then explore the available clinical data to see if any observed clinical data correlates with the known function of the gene involved in the microarray experiment. Finding correlations in this manner could be described as a "hit-or-miss" procedure and is also dependent upon the accumulated knowledge of the researcher. Further, the large volumes of data that are generated by current experimental data generating procedures, such as microarray procedures, for example, ~~makes~~ make this method of correlating an extremely tedious, if not impossible task.

Paragraph [0009] beginning on page 3:

[0009] A tool for forming a compressed view of gene expression results from multiple microarrays is described in co-pending and commonly owned Application Serial No. 10/209,477 filed July 30, 2002 and titled "Method of Identifying Trends, Correlations, and Similarities Among Diverse Biological Data Sets and System for Facilitating Identification", which is incorporated herein in its entirety, by reference thereto. In one example, microarray experimental data used to generate the compressed visualization was obtained from the National Human Genome Research ~~institute~~ Institute of the National Institutes of Health. Experiments were performed with respect to thirty-one subcutaneous melanoma patients using DNA microarrays. For each patient, eight thousand and sixty-six individual microarray measurements were displayed. Additionally, clinical data as well as patient cluster, and gene specific annotations corresponding to the gene represented by the expression ratios were contained within the respective rows of microarray data. Since the data set is highly de- normalized, for a given patient, the data in the clinical columns was repeated for each gene measured by that patient's microarray. In order to display such a massive number of columns in a single visualization,

this system also employed Table Lens, which allowed the diverse data sets to be compressed, displayed and inspected simultaneously in graphical form on a single display. In this example, the system was based on a product known as Eureka, by Inxight. A complete description of the functionality of Table Lens can be found in U.S. Patent Nos. 5,632,009; 5,880,742 and 6,085,202, each of which is incorporated herein, in its entirety, by reference thereto. The resultant visualization was a very dense graphical display representing 241,980 rows of data entirely visible on a single standard computer display. The visualization was highly compressed, with graphical values displayed to represent groups of cell values, since the compression prevented each individual row or cell value from being displayed. The tool further provides the capability of sorting by various data categories, such as "patient cluster" and "invasive ability", for example, as described in the application. As a result of such sorting operations, correlation may be observed between patient clusters, or other categorical criteria. Although the system and methods described in the above system can be very useful and powerful in preparing visualizations for the analysis of biological analysis, they also require a significant amount of learning and familiarization with what is otherwise a quite non-intuitive display for those trained in the biological research disciplines. Those users that have not dedicated enough time to fully understand how to manipulate and interpret the display are likely to be confused or intimidated by the graphical representations of the compressed data and as to how to interpret them.

Paragraph [0011] beginning on page 5:

[0011] The present invention provides systems, methods and recordable media for manipulating large data sets for visually identifying relationships among the data that can be useful to a researcher. By manipulating the data according to the present methods, sorting of the data may be accomplished relative to one or more pseudo-data vectors calculated from any of a variety of sources. Data can be easily and quickly manipulated by sorting or ~~re-ordering~~ reordering rows or columns to expose potentially meaningful correlations and trends in the data which are easily observed.

Paragraph [0012] beginning on page 5:

[0012] A pseudo-data vector may be calculated from data ~~this~~ that is descriptive of the dataset being examined, but not part of the actual data in the dataset. A pseudo-data vector

may be calculated from an entire row (or column) of descriptive data, or even only a portion thereof, for example when one or more data values ~~is~~ are missing from the row or column of descriptive data. User input may be provided for, wherein a user or the system may input predetermined values to be substituted for the descriptive data values.

Paragraph [0032] beginning on page 7:

[0032] Fig. 9B schematically shows ~~shown~~ a classification row 320 having binary values, being converted to a pseudo-experimental vector to be used as a basis for similarity sorting.

Paragraph [0078] beginning on page 13:

[0078] Standard heat map visualizations have significant shortcomings as to their usefulness for performing visual correlation analyses. Since these displays are static, the cells in the display 200 cannot be manipulated to form different combinations or views in attempting to find similarities among the experimental data. Although a commonly owned product, known as Synapsia (available from Agilent, Palo Alto, California) provides some limited capability such as simple column sorting or column rearrangement of a heat map, there remains a need for greater manipulation of the data such as provided by the present invention. Further, as noted above, the sheer volumes of data that are generated by current experimental data generating procedures, such as microarray procedures and protein expression measurements, for example, ~~makes~~ make it generally impossible to display the contents of all the data that needs to be reviewed on a single display. This further complicates any hope for visually identifying similarities among experiments or gene expression values, since not only is ~~side by side~~ side-by-side visualization of potentially similar data values not currently possible through use of an automated technique, but the user must additionally switch between screen views to search for similarities, which eliminates the potential for simultaneous viewing of many of the possible combinations of the data.

Paragraph [0080] beginning on page 14:

[0080] In addition to the experimental data, clinical data and patient data are included in portions 120 and 130 of the visualization 100 adjacent matrix 110 shown in ~~Fig.2~~

Fig. 2. The column 43 labeled "Unigene" contains the Unigene Cluster ID that further identifies the CDNA having been deposited on the microarray, with respect to each of the respective cells in each array 1-31. Thus, for example, Unigene Cluster ID "Hs 23590" is associated with the first row of experimental data 110 as shown in Fig. 2. This identifier is linked to that particular row of array data, so that if the row is reordered within the array, the Unigene Cluster ID is also reordered to the same row that the data assumes, to maintain accuracy of the characterizing clinical data. Likewise, the column of clinical data containing the clonelD (i.e., "Clone") 44 for the CDNA having been deposited on the microarray with respect to each individual microarray reading is linked to the particular row of experimental data that it describes and moves with that row when the row is repositioned. All other columns of clinical data share this characteristic. Columns 46, 48, 50 and 52 contain Name, BNS Symbol, BNS Description, BNS Chr data for each gene having these identification data in its row. The BNS columns 48, 50 and 52 contain information that is all imported from a commonly owned biological naming system, which is described in more detail in co-pending and commonly owned Application Serial No. 10/154,529 filed May 22, 2002 and titled "Biotechnology Information Naming System", which is hereby incorporated in its entirety, by reference thereto. The BNS columns 48, 50 and 52 are only examples of additional descriptive or annotative data that may be displayed along with the experimental data according to the present invention, and the present invention is in no way to be limited to inclusion and use of BNS information in each instance of use of the present invention.

Paragraph [0091] beginning on page 19:

[0091] As noted above, the row sort was performed on the basis of the expression values in row R9 (i.e., Melan-A gene). As each of the cells in row R9 are rearranged according to the sort order determined, the entire column of experimental data assumes the same column placement as that of the reordered cell of row R9. Also, the non-experimental data and identification data in the top portion of the visualization remains linked with the respective columns that it originally pertained to, and is rearranged according to the sort order of the cells in row R9. In this way, the identifying information/ non-experimental data in the cells of rows R1- R4 remains in the same row relative to the

experimental data after ~~re-ordering~~ reordering, thereby maintaining the accuracy of the normalization scheme. The non-experimental data on the left side of the visualization 100 remains locked, as it is normalized with respect to the rows of experimental data, which were not reordered in this manipulation.

Paragraph [0094] beginning on page 20:

[0094] The present invention supports both row and column sorting, as described above, as well as limited column and row ~~re-ordering~~ reordering. This limited column and row ~~re-ordering~~ reordering may be accomplished manually by the user. To accomplish manual reordering, the user can drag-and-drop rows and columns. This is accomplished by simply clicking the column or row header and while holding down the mouse button, dragging it left or right (column) or up or down (row) to its new location.

Paragraph [0097] beginning on page 21:

[0097] Non-experimental data such as that displayed in rows R1- R4 can be loaded in a normalized scheme, in step S3 in an "n x y" matrix, where "n" is a positive integer representing the number of columns in the matrix, which will be displayed as an extension of the columns displaying the experimental values of the n x m matrix, and ~~"y'~~ "y" is a positive integer representing the number of rows in the matrix. The "n value" (i.e., n = 1, 2, 3...n) of each column of the n x y matrix is linked to the corresponding "n value" in the n x m matrix in step S5, so that when a column of the experimental data is reordered by a sort, the column in the n x y matrix which corresponds to the column of experimental data that is reordered is reordered along with it to maintain the proper identification of each column of experimental data by the correct non-experimental data. This linking may be accomplished via BNS-like mechanisms that can match up identifier schemes (even when they are different, as long as a mapping between them exists). In some simple cases the identifiers may be consistent between the two data sets and it is only required that the identifier column is known. This may be by convention (e.g., the first column of every table must be a gene identifier derived from Unigene). Another way of accomplishing the linking ~~it~~ is to require the user to identify the column to be used for linking, at the time that the data is imported for use by the present system in creating a display and manipulating the data displayed therein. Still another technique for

linking is to program the software to analyze the data as it is imported and determine if a column contains recognizable identifiers. For example, the system may scan all the data during import and determine that all entries in a particular column have a recognizable identifier (e.g., all entries in column two start with "Hs.") and so are probably Unigene identifiers and can be used to accomplish the linking. Another example is that all entries may start with "NM_" and so are refseq mRNA identifiers, which can be used as a basis for the linking. Although the last technique described is highly domain specific, it provides useful functionality for users in that domain.

Paragraph [00100] beginning on page 22:

[00100] After constructing the underlying matrix as described above, which serves as the basis for displaying the visualization 100, the data from the matrix is displayed in a single visualization made up of a k x j matrix (step S13, Fig. 5B). The k x j matrix will generally be limited by the capacity of the monitor or display upon which the visualization is outputted, and may be predetermined by the display software. It is generally preferable to display as much data as can be reasonably viewed by the user without over-taxing the eyesight, and it is generally preferable, although not absolutely necessary, to display all of the non-experimental data and all of the columns of the experimental data, so that, for example, in Figs. 2-4, at least a portion of the data from each microarray is visible. According to this preference, "k" would be a positive integer equal to the sum of "n" and "z", i.e., k = n + z. Note that some or all of the non-experimental data may need to be abbreviated or cut off, but a tooltips feature may be provided so that when a user hovers the mouse sprite over a compressed, abbreviated or cut-off representation of non-experimental data in a cell, a pop-up display of the full expression of the non-experimental data is displayed. Also, if "n + z" is a value greater than a preset maximum value for "k", then some of the columns of the experimental data may not be displayed, although these values will still be considered in performing manipulations and they may be displayed upon reordering of the columns of experimental data. As to the number of rows displayed in the visualization, the display will be generally inadequate to display all of the rows in examples where the experimental data represented is microarray data or protein abundance data for example. In these instances "j" is an integer equal to the number of rows that

can be reasonably visualized on the display and can be preset in the software, but will be less than the sum of "m + y". Generally, the system is arranged so that all of the rows of non-experimental data ~~is~~ are displayed, while only a first portion of the "m" rows of experimental data is displayed. The experimental data and non-experimental data in rows higher than "j" are accessible by the manipulations of the data, but will only be displayed upon reordering, when one or more rows of the experimental data has been determined by a sort to be of particular interest. The situation where not all columns of experimental data can be displayed does not occur as frequently as the situation when not all the rows may be displayed. For example, when considering microarray data, each column pertains to a microarray and the number of microarrays to be considered can be easily controlled by the user.

Paragraph [00101] beginning on page 23:

[00101] Upon viewing the display 100, if the user decides to perform a column sort at step S15, then the user outlines a row of the experimental data display 110 in step S17 (i.e., the a^{th} row of the total "m" number of rows, where "a" can be any integer from "1" to "j" of the experimental data) which contains data of interest upon which the user desires to perform the column sort. The outlining may be accomplished by aligning the cross hair 114 as described above, or by other visual indicating means. Upon selecting the a^{th} row, as described, each experimental data value (i.e., cells one through n of the a^{th} row, noted as cells 1,a through n,a in step S19)~~)~~ are compared to perform a new sorting order, whether the cells are to be arranged in descending order of value or ascending order of value. This sorting schema is an iterative process in which the first cell is compared with the second to determine the sorting arrangement and then either the first or second cell, whichever is determined to be of lower value according to the sorting schema is compared with the value of the third cell, and so forth, and can readily be accomplished by one of ordinary skill in the art. It is important to note, however, that cells one through z of the a^{th} row of the z x m matrix are not considered or compared during the sorting procedure, as they contain non-experimental data that would be meaningless or erroneous to compare with the experimental data values during the sort.

Paragraph [00106] beginning on page 25:

[00106] The column, row and manual sorting procedures described above can be useful in identifying correlations, trends and other relationships among the data in some instances. However, when dealing with large volumes of experimental data, such as microarray data sets or protein or other molecular data sets, the data sets are often sufficiently "noisy" that it is often difficult to find meaningful correlations by simply sorting a single column (e.g., a single array) or a single row (e.g., a single gene). When experimental data such as these are measured by very low level signals, there may be a lot variation in the measured values from experiment to experiment and they are inherently ~~"noisy'~~ "noisy". Microarrays are generally noisy due to a number of experimental variances. Microarrays are generally qualitatively reproducible, but the individual measurements will still show quite a bit of variance. Thus, if a sort is performed on the basis of a single or individual array, slightly different ordering results are observed, as compared to the same sort performed on an array which is already known to be similar. These differences may even occur when a sorting procedure is performed on two different arrays representing the same experiment (i.e., a replicated experiment) due to differences in noise levels between the two arrays. To address these problems, the present invention further provides the capability of performing similarity sorting, which includes the ability to sort the data set by row or column similarity.

Paragraph [00108] beginning on page 26:

[001081 Fig 6A shows a simple 3 x 4 matrix which will be used to refer to a very simple demonstration of similarity sorting according to the present invention. The 3 x 4 matrix represents ~~and~~ an experimental data set, i.e., an "m x n" matrix as described above with regard to Figs. 5A-5B. Of course, the actual experimental data sets which will generally be treated by the present system and methods will be much larger, such as the 31 x 8,066 matrix referred to in the examples above, but a 3 x 4 matrix has been shown to greatly simplify an explanation of the procedures, while at the same time, explaining the concepts and techniques required, which can then be readily applied to larger data sets.

Paragraph [00109] beginning on page 27:

[00109] A similarity column sort or similarity row sort may be performed on any

of the columns (101, 102, 103) or rows (201, 202, 203, 204) that the user so selects. Thus, for example, assume a user wishes to perform a similarity sort on row 202. By selecting row 202 in Fig. 6A, such as by using the cross hair 114 or other indication means, such as by right clicking on a column or row header or cell representing an experimental data value, the system invokes a popup menu 180, as shown in Fig. 6B. Popup menu 180 gives the user options, among others, of performing a standard sort or a similarity sort. In the view shown in Fig. 6B, a similarity sort has been selected, and the system at this time provides further options as to whether the similarity sort is to be performed according to the current row selected 185 or current column selected 186. Although not shown, selection of a standard sort would provide the same options (i.e., as to row or column based sorting), and sub-sorting as well as next neighbor sorting options may also be provided in the popup menu 180 or a similar popup feature. After selecting a similarity row sort in this example, the system rearranges the matrix of experimental data such that row 202 becomes the first row positioned in the matrix as shown in Fig. 6C. Any non-experimental data (e.g., data in the z x m matrix characterizing rows 201 and 202 (which happen to be the only two rows that were repositioned at this stage) is repositioned so as to maintain the positions relative to the experimental data prior to the row reordering.

Paragraph [00114] beginning on page 29:

[00114] The Euclidean measurement technique described may be desirable for finding rows (or columns) which are closely similar in overall amplitude, while the Pearson correlation coefficient may be more desirable for sorting ~~a~~ and separating correlated and anti-correlated rows (or columns), though similarity in this approach is weighted more toward the overall pattern or shape of an expression profile, rather than its amplitude. In any case, the user may select among similarity measurements and may choose to approach the data with more than one type of similarity measurement, to compare and contrast the results achieved.

Paragraph [00122] beginning on page 32:

[00122] It is further noted that the similarity sorting procedures described above are only one approach to reordering data based on similarity among entire rows or columns of data. Various other approaches to manipulating the experimental data based upon characteristics of entire rows or columns may be readily applied by the instant invention. As just one further

example, a similarity sorting order can be computed to group "nearest neighbors" of rows or columns. According to this approach, the selected row or column is positioned first followed by the row or column with the shortest squared Euclidean distance or other lowest valued sorting criteria (i.e., nearest neighbor). The third row or column is selected based on its determination as the nearest neighbor to the second row or column and positioned adjacent thereto, and so forth. According to this procedure, all rows or columns are calculated for similarity or proximity to the selected (first positioned) row or column, just as in the above-described procedure, to determine positioning of the second row or column. However, this approach varies for placement of the third and subsequent rows/columns. For the second and subsequent row/column positions, the distance/proximity calculations are repeated or iterated wherein the row/column positioned just filled is treated as the selected row/column. For example, for placement of the third row/column, the second placed row or column is used to ~~determined~~ determine distances/proximities with respect to all remaining rows/columns except the first row/column which has already been placed. By this iterative treatment of the data, what results is an ordering wherein the second row/column is the nearest neighbor of the first row/column; the third row/column is the nearest neighbor of the second row/column; the fourth row/column is the nearest neighbor of the third row/column, and so forth, as contrasted with the previously described procedures where each row/column is ordered based upon its relative similarity to the first column/row. By this approach, each adjacent row/column is positioned so as to be relatively similar to its neighbors and this provides an additional view by which the user might identify emerging trends among the experimental data.

Paragraph [00128] beginning on page 35:

[00128] The procedures and techniques described above with regard to similarity sorting of the experimental data may also be applied to non-experimental data, to provide similarity sorts based on the non-experimental data that provide insights to similarities between various rows or collections of the experimental data. In many cases the non-experimental data, which accompanies and describes the experimental data ~~this~~ that is displayed, may be represented by a binary set of values, for example, "yes/no", "true/false", "male/female" , "+/-", etc. For example, a row may be provided to characterize whether the samples from which the

experimental data are taken are diseased or not (which may be represented by "yes/no" or "+/-", for example) or whether the samples have been drug treated or not (the values of this row may also be represented as "yes" or "no"; or "+" or "- ", for example), or whether the sample is taken from a female or male (values of this row may be represented by "F" or "M", for example). These are only examples of non-experimental data meeting the binary criteria and are in no way limiting of the present invention, as there are many more categories of information that may be used. The classification of such non-experimental data may be clinical, phenotypical, computational (e.g., partitions derived computationally, see Bittner, M et al., "Molecular Classification of Cutaneous Malignant Melanomas by Gene Expression Profiling", which was incorporated by reference above), or other descriptive data characterizing the data in matrix 110. Further, classification data may even be experimental data that is not a member of the set of experimental data included in matrix 110 (e.g., experimental data describing the experimental data in matrix 110).

Paragraph [00130] beginning on page 36:

[00130] In order to perform similarity sorting against a selected row of data that is classified according to a binary classification scheme (such as any of rows R2, R4, R5 and R6 noted above), a user selects a row of binary classified classification data as a row of interest to serve as a basis for the sort procedure. The system then produces an imaginary row of expression data, also referred to as a row of pseudo-experimental data, based on user settings for values to be applied to the existing binary values. Fig. 9A shows an example of a menu item provided the user for setting the pseudo-values of the binary data according to the user's preferences. In the example shown, the user has selected row R6 ("Cluster per H...) as the row upon which to perform the similarity sort. Menu 300 provides a selection 302 for setting the positive value of the binary values, as well as a selection 304 for setting the negative value. In this example, the user has assigned an 8.0 fold increase value to the positive value (i.e., "+", in this example), and an ~~8~~ 8.0 fold decrease (i.e., 1/8.0) value to the negative binary value (i.e., "-", in this example). The positive and negative values are settable by the user so that if the user wants to create a pseudo-experimental vector with relatively extreme amplitudes, the negative and positive values can be ~~sent~~ set

relatively high. On the other hand, the user may choose lower values to create a vector with lesser amplitude swings. Any values that are non-reported (i.e., neither "+" nor "-", in this example) may be automatically assigned a null value, which is a value of one, for purposes of gene expression ratio measurements, since they are generally normalized log ratios. Thus, an expression ratio of one corresponds to no up or down regulation. For datasets that are not characterized by ratio values or log ratio values, however, the system may substitute a null value of zero.

Paragraph [00133] beginning on page 38:

[00133] It is further noted that meaningful similarity sorts have been successfully performed even upon incomplete information in a row of non-experimental data. For example, a similarity sort performed after converting row R2 of Fig. 8 produced results qualitatively similar to those produced computationally by Bittner et al. (see Bittner, M. et al., "Molecular Classification of Cutaneous Malignant Melanoma by Gene Expression Profiling", referred to and incorporated by reference above), in that the same highly discriminating genes were identified from the overall dataset, as being significant to the sort that was carried out. Thus, even though the binary information for "Vasculogenic mimicry" (row R2) is only known for some cell lines, the conversion to a pseudo-experimental vector substituted values corresponding to ratios of one for the unknown values, as noted. The resulting vector still contained sufficient information to identify relevant genes. It is further noted that the current software may be set to toggle between assignments of the user set positive and negative values, so that the order inverts each time a sort is run. For example, when set in toggle mode, a first similarity sort may produce a pseudo-experimental data vector by assignment of the positive set value to "+" values in a class of binary data and by assignment of the negative set value to the "-" values in the binary data. Then, on the next successive search, the system assigns the positive set value to the "-" binary values and assigns the negative set value to the "+" binary values in the binary ~~data..~~ data. This enables sorting both possible constructions of the pseudo-expression vector with the same simple user interface.

Paragraph [00136] beginning on page 40:

[00136] The system allows a user to identify a select group of cells (e.g., a group of cells from a row of experimental data, from which a pseudo-experimental data vector is then generated for use as a basis for similarity sorting. This type of search may be useful in an instance, for example, where the user knows that certain particular columns in the matrix 100 identify samples ~~know~~ known to be important to a process being studied, for example, a group of columns may be tissues taken from a tumor registry and the experiments may be studying a particular type of cancer. In this case, the cells aligned with the columns ~~identify~~ identifying tumor registry samples are likely to be ~~effected~~ affected or upregulated upon occurrences of the particular cancer being researched. Therefore, by searching those rows that distinguish the selected columns from the remaining columns of the experimental data, this is likely to find a cluster of related expression data vectors. It should be noted here that the cells selected for creation of a pseudo-experimental data vector are typically contiguous cells in a row of experimental data, although they need not be. The same techniques can be carried out on noncontiguous cells in a row of experimental data, contiguous cells in a column of experimental data, or non-contiguous cells in a column of experimental data.

Paragraph [00149] beginning on page 44:

[00149] Next, at event 1212, it is determined whether there are remaining rows of data, which have not been reordered (i.e., unsorted rows), upon which to carry out an additional similarity search. If there are no unsorted rows remaining, then the process ends at event 1214. If there are unsorted rows remaining (i.e., those that have not been already displayed as ordered sort results), then it is determined at event 1216 whether there are any columns remaining which have not been selected by the window as it is incremented, i.e., is the product of the number of increments times the increment size 308 less than the total number of columns? If the product of the increment number times the increment size is not less than the total number of columns, then the process ends at event 1218. If on the other hand, the product is less than the total number of columns, then the counter is incremented by one at event 1220.